Pergamon

Pattern Recognition, Vol. 27, No. 7, pp. 1039–1049, 1994
Printed in Great Britain

0031-3203/94$6.00 + .20

0031-3203(94)E0019-H

(3,4)-WEIGHTED SKELETON DECOMPOSITION FOR PATTERN REPRESENTATION AND DESCRIPTION

GABRIELLA SANNITI DI BAJA† and EDOUARD THIEL‡

†Istituto di Cibernetica, CNR, Via Toiano 6, 80072 Arco Felice, Naples, Italy
‡Equipe TIMC-IMAG, CERMO BP 53, 38041 Grenoble Cedex 9, France

(Received 29 May 1992; received for publication 24 February 1994)

Abstract—A digital pattern, perceived as the superposition of elongated regions, is decomposed into simple regions through the decomposition of its (3,4)-weighted skeleton. The skeleton is interpreted as a curve in 3D space, where the three coordinates of any pixel are its planar coordinates and the distance label. The 3D curve is divided into rectilinear segments, which constitute the spines of elementary regions, i.e. regions with linearly changing width and orientation. Then, the spines are analysed to simplify the skeleton decomposition and avoid redundancy. Spines identifying regions unnecessary for the description of the pattern are annihilated, while contiguous spines, corresponding to sufficiently similar regions, are merged. The resulting skeleton components are used to represent and describe the simple regions into which the pattern is decomposed. Decomposition at different resolution levels can be obtained by selecting different threshold values during the polygonal approximation, performed to divide the skeleton into rectilinear pieces, and/or the successive merging step.

Weighted distance Labelled skeleton Polygonal approximation Decomposition Description

1. INTRODUCTION

The description of patterns that can be perceived as the superposition of ribbon-like regions[1] can be facilitated by the structural approach. A suitable representation of the pattern is decomposed, in such a way that each decomposition component could be interpreted as the representation of one of the regions constituting the pattern.[2–7] Then, the description of the pattern is obtained in terms of the description of the obtained regions and of their spatial relationships. The description of each elementary region, which by hypothesis is characterised by a simple shape, can be obtained by exploiting the information carried on by the corresponding component of the representation system.

The labelled skeleton[8–13] is a convenient tool to analyse the shape of patterns perceived as union of ribbon-like regions. A ribbon-like pattern is characterised by one spine and a disc: the disc sweeps out the shape by moving along the spine, changing size as it moves. The skeleton is a curvilinear subset of the pattern and its branches play the role of the spines of the ribbon-like regions constituting the pattern. The label of any pixel p of the skeleton, which represents the distance of p from the complement of the pattern, can be interpreted as the radius of the sweeping disc centred on p. The shape of the disc depends on the adopted distance function. The discs are more rounded if a quasi Euclidean metric is adopted.

Reasonable approximations of the Euclidean distance are provided by the weighted distance functions,[14–17] where suitable integer weights are used to measure the distance between neighbouring pixels, depending on their relative position. Skeletons[3,10–12,18] whose pixels are labelled using a weighted distance are called weighted skeletons. The weighted skeleton and the skeleton labelled according to the city-block or the chessboard distance can be obtained at a comparable (limited) computational cost. The stability of the weighted skeletons under pattern rotation favours their use for practical applications.

A correspondence exists between any subset of the skeleton and the region of the pattern that is the union of the discs associated with the pixels of the skeleton subset (in the strict sense, the only pixels of the skeleton subset which are centres of maximal discs are enough to recover the region). This region can be obtained by applying the reverse distance transformation[18,19] to the skeleton subset, which requires two raster scan inspections when a sequential algorithm is used. Under certain circumstances, a satisfactory approximated version of the region can be obtained at a lower computational cost. For instance, if the skeleton subset can be interpreted as the spine of an elementary region having linearly (and monotonically) changing width and orientation, a satisfactory approximated region is the envelope of only two discs, those associated with the extremes of the spine.

In this paper, we divide the (3,4)-weighted skeleton[7,13] of a pattern into subsets that can be understood as spines of simple regions. The decomposition method has been inspired by previous works[19,20] where the city-block distance labelled skeleton has been employed. Skeleton decomposition is accomplished in two main

skeleton branches. Pruning is iterated until branch removal does not diminish the representative power of the skeleton, in the limits of the adopted tolerance. Thus, skeleton branches which are initially delimited by branch points may be pruned as soon as they become peripheral branches, due to the deletion of the neighbouring branches. Iterating pruning does not cause a summation effect in the loss of information. At each iteration and for each peripheral skeleton branch, the protrusion whose relevance is evaluated is the protrusion mapped in the union of the current peripheral skeleton branch with the neighbouring skeleton branches, already pruned at a previous iteration. To this purpose, the information relative to the starting point(s) of the branch(es) is propagated through the branch(es) while performing pruning.

In Fig. 3, the (3,4)-weighted skeleton is shown superimposed on the input pattern, before and after pruning. Pixels denoted by "+" are not recovered when the reverse distance transformation is applied to the skeleton. In Fig. 3(a), only a few pixels of the border of B are missed out. Loss of recovery happens since the skeleton is required to be unit-wide and, as such, it does not include all the maximal centres of B. In Fig. 3(b), the smoothing effect due to the pruning process is evident; each skeleton branch remaining after pruning corresponds to a significant ribbon-like region.

This skeleton is preliminarily decomposed into the constituent skeleton branches. This is equivalent to performing a decomposition of the pattern into the elongated regions that could be obtained by individually applying the reverse distance transformation to the skeleton branches. A data structure is built to record the extremes of the skeleton branches and the spatial relationships among them.

Each skeleton branch is furthermore decomposed, by means of a polygonal approximation, in such a way that each rectilinear segment constitutes the spine of an elementary region. Division points have to be placed wherever non-linear curvature changes occur along the skeleton branch, as they reflect non-linear curvature changes along the contour of the corresponding pattern subset. Division points have also to be placed where non-linear or non-monotonic label variations occur, as they indicate non-linear or non-monotonic pattern thickness variations. To locate both types of division points, we interpret any skeleton branch as an arc in 3D space where, for each skeletal pixel, the three coordinates are the planar coordinates and the normalised label. Using the normalised label in place of the distance label is done to treat uniformly the three coordinates, by allowing a displacement of one unit only in each of the three directions, when passing from a skeletal pixel to one of its neighbours. In this way the skeleton branch is a connected arc also in the 3D representation.

The polygonal approximation is accomplished by using a split type algorithm (e.g. the one described by Pavlidis[(14)]), so that the obtained set of vertices is not influenced by the order in which skeletal pixels are processed. The extremes of the current branch (say v_i and v_j) are accepted as vertices, and properly stored in the data structure. Then, new vertices are identified (and stored in the data structure) in a recursive way. The Euclidean distance $d_E(p)$ between any pixel p of the skeleton branch and the 3D straight line (v_i, v_j) is computed. Then, the pixel of the branch for which $d_E(p)$ has the largest value is taken as a new vertex v, provided that $d_E(p)$ is greater than an a priori fixed threshold θ. Vertex selection is then accomplished on the sub-arcs $v_i v$ and $v v_j$. The recursive process terminates when, for the pixel maximising $d_E(p)$, it results in $d_E(p) \le \theta$.

If a number of pixels of the skeleton branch maximises $d_E(p)$, the two pixels which are respectively the closest to v_i and to v_j are accepted as vertices, and the skeleton arc is divided in three sub-arcs, which are recursively examined. Accepting as vertices all the pixels maximising $d_E(p)$ could result in a polygonal approximation with too many vertices, not necessarily all significant. In contrast, accepting only one pixel could make the skeleton decomposition dependent on the order in which the pixels are processed.

The square root computation necessary to obtain the Euclidean distance $d_E(p)$ can be avoided, since the same result is obtained when computing the square distance with the square threshold.

The value of θ is fixed depending on the tolerance regarded as acceptable for the specific task. The threshold should be rather small, to favour a quite faithful recovery of the elementary regions having the skeleton segments as their spines, by building the envelopes of the pairs of discs centred on the extremes of the spines. In our experiments, the value $\theta = 1.5$ has been revealed as adequate. Skeleton decomposition at different resolution levels is obtained by assigning different values to θ. As the threshold increases, the number of components into which the skeleton is decomposed generally diminishes, while the representation becomes rougher and rougher. In fact, the regions that could be recovered by applying the reverse distance transformation to the pixels of the skeleton components are likely to differ remarkably from the envelopes of the discs centred on the extremes of the so-found skeleton components.

Let $V(\theta)$ be the set of vertices found in the polygonal approximation of the skeleton, performed with the lowest threshold. The vertices of any other polygonal approximation, performed with a higher threshold, can be directly identified as they constitute a subset of $V(\theta)$. Any subset is obtained by comparing the value $d_E(p)$, stored for any pixel p of $V(\theta)$, with the desired new threshold.

Recovery of the elementary regions represented by the skeleton segments is not necessary for computing geometric features (e.g. area or perimeter) and shape features (e.g. orientation or rectangularity) of the regions. These features can easily be derived starting from the 3D coordinates of the found vertices. However, for illustrative purpose, the elementary regions corresponding to polygonal approximations of the skeleton at three different threshold values, are shown in Fig. 4.

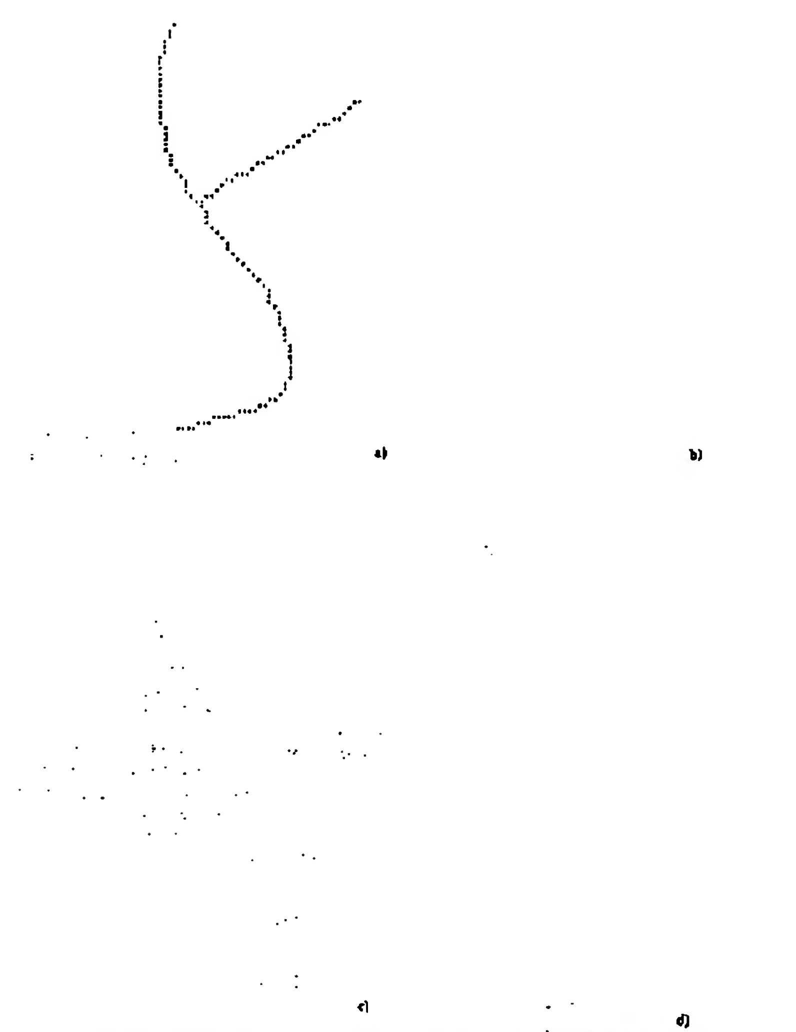


Fig. 4. Letter V denotes the vertices found during the polygonal approximation of the skeleton with threshold $\theta = 1.5$ (a); decomposition into elementary regions, corresponding to $\theta = 1.5$ (b), $\theta = 4$ (c), and $\theta = 8$ (d).

To use all the pattern representations in a compact way, we associate each vertex p of $V(\theta)$ a quadruplet (x, y, label, $d_E(p)$). In this way, the permanence of a vertex in any of the resolution levels can be immediately checked. In Table 1, the entries x, y, l, d, and t indicate the Cartesian coordinates x and y, the label, the 3D distance, and the pixel type, respectively (b, e, and n, stand for branch point, end point and normal point.

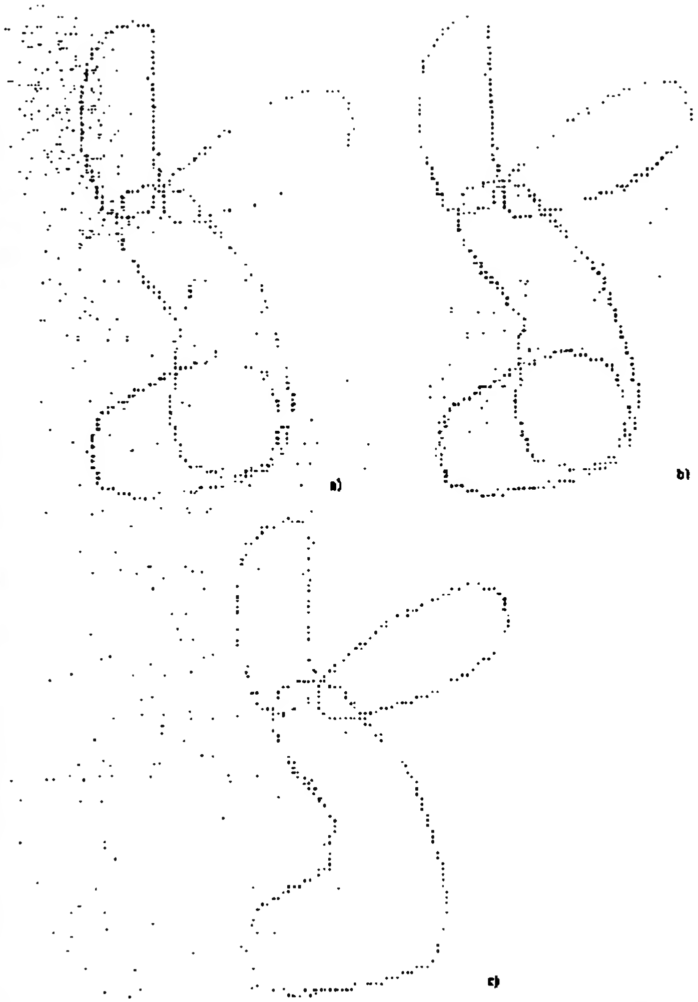


Fig. 7. Different decompositions of the same pattern, obtained by using different values for the merging threshold: $\tau = 0.15$ (a), $\tau = 0.25$ (b), $\tau = 0.50$ (c).

Each pair of successive spines, belonging to the same skeleton branch, is examined. Let (v_{i-1}, v_i) and (v_i, v_{i+1}) be the vertices delimiting the current pair. Let D_i and L_i be the Euclidean distance of v_i from the straight line segment joining v_{i-1} and v_{i+1}, and the Euclidean length of the segment, respectively. A flag F, initially equal to 0, is set to 1 in correspondence with each vertex v_i such that D_i/L_i is less than an a priori fixed merging threshold τ.

Let $v_1, v_2, \ldots, v_n$ be a set of successive vertices, in correspondence of which it is $F = 1$. Moreover, let v_0 and v_{n+1} be the vertices immediately preceding v_1 and immediately following v_n.

If $n = 1$, the two spines (v_0, v_1) and (v_1, v_{n+1}) are merged by all means.

If $n > 1$, the distance D_i from the straight line segment joining v_0 with v_{n+1} is divided by the length L_i of the segment, for every v_i ($i = 1, 2, \ldots, n$). If $D_i/L_i < \tau$ for every vertex, all the spines are merged. Otherwise, the concatenation $v_1, v_2, \ldots, v_{n-1}$ is considered, and for each of these vertices the merging ratio D_i/L_i is checked with reference to the straight line segment joining v_1 and v_n. The process is repeated until for the concatenation $v_k, v_{k+1}, \ldots, v_j$ ($k = 1 + i$, $j = n - i$, $i > 0$) the merging condition is verified by all the vertices. Then, the merging condition is recursively checked on the two sub-concatenations $v_1, v_2, \ldots, v_{k-1}$ and $v_{j+1}, v_{j+2}, \ldots, v_n$.

The vertices delimiting the set of the merged successive spines are taken as the extremes of the resulting complex spine. Note that the remaining vertices still maintain their region representation power, since the region associated with a complex spine is the union of the elementary regions associated with the merged spines.

The value of the merging threshold τ depends on the desired merging tolerance. In our experiments, the value $\tau = 0.25$ has been adopted as a default value. Larger values can be used to favour merging. An example is shown in Fig. 7, where three different values have been used for the merging threshold τ. The three decompositions are obtained starting from the polygonal approximation of the skeleton, performed with $\theta = 1.5$. Note that, in contrast to the decompositions shown in Fig. 4, the regions are not elementary regions.

The possibility of merging spines sharing a branch point as a common vertex could also be taken into account, so that the final pattern decomposition would not be conditioned by the preliminary decomposition of the skeleton into its constituting branches. Work in this respect is currently in progress.

5. CONCLUSION

In this paper we have illustrated a method for decomposing a digital pattern through the decomposition of its weighted skeleton. The method is adequate for patterns that can be perceived as constituted by the union of elongated (ribbon-like) regions. It could be employed, for instance, in the framework of a document analysis task to classify the alphanumeric symbols which it contains.

The weighted skeleton has been chosen to favour the stability of the decomposition under pattern rotation. In fact, stability is an indispensable presupposition for

Fig. 8. Stability of the decomposition under pattern rotation.

References

[1] A. Y. Commike and J. J. Hull. Rule learning for syntactic pattern recognition. In *Proc. 4th U.S. Postal Service Advanced Technology Conf.*, pages 621–633, November 1990.

[2] J. Franke. Experiments on the cenparmi data set with different structured classifiers. Nov. 30 – Dec. 2 1992. To appear in *Proc. 5th U.S. Postal Service Advanced Technology Conf.*

[3] J. Franke. On the functional classifier. In *Proc. 1st Int. Conf. on Document Analysis and Recognition*, pages 481–489, St-Malo, France, September 1991.

[4] P. D. Gader, D. Hepp, B. Forester, T. Peurach, and B. T. Mitchell. Pipelined systems for recognition of handwritten digits in USPS zip codes. In *Proc. 4th U.S. Postal Service Advanced Technology Conf.*, pages 539–548, November 1990.

[5] J. J. Hull, A. Commike, and T.-K. Ho. Multiple algorithms for handwritten character recognition. In *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pages 117–124, Concordia University, Montreal, April 1990.

[6] K. Kimura and M. Shridhar. Handwritten numeral recognition based on multiple algorithms. *Pattern Recognition*, 24(10):969–983, 1991.

[7] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, and H. S. Baird. Constrained neural network for unconstrained handwritten digit recognition. In *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pages 145–154, Concordia University, Montreal, April 1990.

[8] R. Legault and C. Y. Suen. A comparison of methods of extracting curvature features. In *Proc. 11th Int. Conf. on Pattern Recognition*, pages 134–138, The Hague, The Netherlands, Aug.–Sept. 1992.

[9] R. Legault and C. Y. Suen. *A Contour-Based Recognition System for Totally Unconstrained Handwritten Numerals.* Technical Report, Concordia University, Montreal, 1989.

[10] H. Nishida and S. Mori. An approach to automatic construction of structural models for character recognition. In *Proc. 1st Int. Conf. on Document Analysis and Recognition*, pages 231–241, St-Malo, France, September 1991.

[11] C. Y. Suen, R. Legault, C. Nadal, M. Cheriet, and L. Lam. Stretching the limits of handwriting recognition systems. In *Proc. 2nd IPTP Conf. on Postal Processing Systems in the 21st Century and Character Recognition*, pages 97–109, Tokyo, Japan, January 1992.

[12] C. Y. Suen, C. Nadal, R. Legault, T. A. Mai, and L. Lam. Computer recognition of unconstrained handwritten numerals. *Proceedings of the IEEE, Special issue on Optical Character Recognition*, 80:1162–1180, July 1992.

The Interpretation and Reconstruction of Interfering Strokes

David S. Doermann and Azriel Rosenfeld
Document Processing Group, Center for Automation Research
University of Maryland, College Park, MD 20742-3411

Abstract

This paper addresses the problem of the interpretation of interfering contours by using perceptual features of the domain to reconstruct intersecting regions. By treating the strokes as features and retaining a more detailed representation of the document, we can use criteria and cues for interpretation which are not available from traditional approaches to document processing.

1 Introduction

In many document understanding domains, complications arise when handwritten strokes interact with themselves or with other markings on the document. If such interferences can be detected, it would then be useful to recover the strokes and markings which participate in the interference, so that subsequent analysis and recognition algorithms will have the benefit of a more complete representation of these features. One promising approach to interpreting such interactions is to use properties of the affected strokes or markings as well as other knowledge about the domain to separate the interfering features.

In our research we address the problem of the interpretation and reconstructing handwritten strokes and machine-produced line segments whose structure has been affected by interaction either with each other or with other document features. Handwritten strokes may cross, merge, or otherwise touch; our goal is to recover the instrument trajectory which gave rise to the features and the properties of the corrupted segments.

The extrapolation of a thin-line stroke representation into singular regions (i.e. junctions) based on contours has been addressed by Nishida and Mori ([3]). Our work is based on an intensity image representation and extends the analysis to general stroke/feature intersections. Reconstruction has also been addressed by Wang and Srihari, who describe an approach to character splicing based on cut points between a line and a character ([4]).

2 Approach

Our approach to the problem of interfering contours is based on the detection, analysis and detailed representation of the stroke-like and non-stroke-like regions in the document image. The process involves two parts: an interpretation of the intersecting region and a reconstruction of the strokes or line segments which formed it.

An interpretation of a region is derived from the local configuration of stroke segments and from properties of the strokes themselves such as curvature, width and intensity. Our analysis relies on the concept of a stroke recovery platform to provide a comprehensive and adaptable representation of the document and is described briefly in Section 3. In Section 4 we discuss the example of a handwritten stroke intersecting a single isolated line segment. In Section 5 we provide a classification of general stroke interactions (e.g. merges,

The support of this research by the Ricoh Corporation is gratefully acknowledged.

crossings, abutments, etc.). We also provide a method of interpretating and ultimately reconstructing the intersection region using the existence of uncorrupted boundaries, smoothness constraints, and properties of the incoming stroke segments.

3 The Stroke Recovery Platform

The framework used to address the interpretation and reconstruction problems is based on the concept of a stroke recovery platform, which is described in [1, 2]. The platform provides a hierarchical representation of the stroke-like features in a document, extending from the pixel level up through an attributed stroke graph.

Figure 1 shows a subset of the features computed for each region of the image. The platform contains, most importantly, a set of cross-section groupings which exhibit stroke-like properties (hypothesized stroke segments), regions which are classified as possible junctions or endpoints, and the underlying contours or contour fragments of the stroke segments, junctions, endpoints and unclassified features in the image. The stroke graph supports top-down access to the pixels through the junctions, strokes, cross-sections and retinotopic (pixel-level) information.

4 Stroke/Line Segment Interaction

To illustrate how the stroke platform can be used to address feature interaction problems we consider the intersection of a handwritten stroke with a straight, machine-produced line segment. This example is sufficient to illustrate the basic problems and serves as a basis for handling recovery tasks involving more complex interactions. Our goals are to isolate the areas which result from a superposition of presumably independent markings and to delineate the original contours of the stroke and the line.

After we have constructed the stroke platform (Figure 1) we identify those portions of the image which correspond to interacting features. If properties of the line segment features such as width, position and orientation are known a priori, the stroke graph can be examined and the features identified. More realistically, we may identify line segments based on the regularity and size of the cross-sections comprising the hypothesized stroke segments. In either case, if the segment intersects another feature, we will find a node in the stroke graph corresponding to the intersection. If the intersection occurs over an extended region, the affected portions of the stroke graph will have cross-section widths which are inconsistent with the rest of the line segment. In Figure 1b, for example, the top-center stroke segment is bounded by two apparent junctions and has cross sections of significantly greater width than the corresponding left or right end segments. Since such changes contradict the assumption of consistency, such a situation should be examined for possible interpretation as resulting from an interaction of features.

Once we have an approximate delineation of the line segment, we begin the reconstruction. As stated earlier, the reconstruction is based on properties of the portions of the document that do not involve feature interactions. We first identify *anchor points* which are used to connect the reconstructed feature segments to known feature segments. We identify a set of candidate anchor point pairs from the cross-sections at the ends of the affected segments (Figure 2). Since the line segment is of known dimensions, we generate (or retrieve as part of our a priori knowledge) a cross section representation of the model line segment and register it with the representation given by the platform. Figure 3 shows a set of ideal cross sections

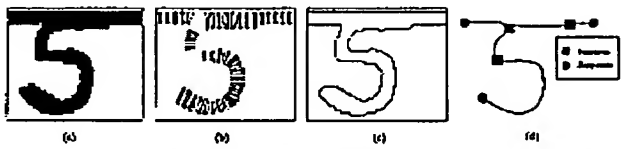


Figure 1: Several components of the platform which result from the processing of the document region (a).

Figure 2: The candidate anchor points derived from the cross section endpoints.

overlayed on the image. From this correspondence, we can easily identify the isolated line segment features which are uncorrupted and refine the registration if necessary. We then classify contours between the anchor points of the hypothesized stroke segments which do and do not fit the model.

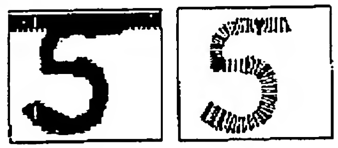
Bounded portions of the contours are described as follows. A *visible* contour is a boundary representation of a stroke or region that is derivable from areas of high gradient activity in the image. An *occluded* contour is a boundary of a stroke or line segment which is obscured or otherwise distorted by another stroke or line segment. A contour is said to be *stable* if it corresponds to an uncorrupted portion of the stroke and is itself free from distortion caused by noise in the intensity image. A contour is *unstable* if its location or orientation may be corrupted by neighboring strokes.

The platform can thus be annotated to reflect line-segment, non-line-segment and possible-combination cross sections, contours and stroke graph components. If necessary the occluded line segment contour is easily recovered from the model by generating its location and position from the visible/stable contour.

The occluded stroke segment contour is then reconstructed from the remaining visible contours and anchor points. For the intersection in this example, the visible stroke contour is connected to the remaining part of the stroke segment, so we can assume that it is part of the same stroke. We use the properties of the unoccluded stroke segments to reconstruct the contour and delineate the region which corresponds to the occluded stroke (Figure 3b).

5 General Stroke/Stroke Interaction

In this section we describe in more detail the stroke extrapolation and reconstruction algorithms. Without loss of generality, it is sufficient to discuss only the interaction among



(a) (b)

Figure 3: The reconstructed line segment in registration with the original image and the stroke recovered from the uncorrupted boundary

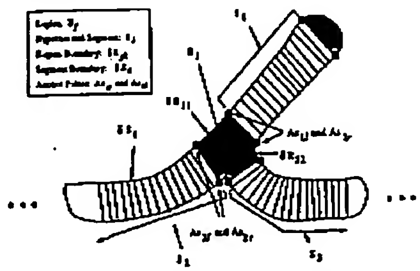


Figure 4: Landmark features for stroke reconstruction

hand-produced stroke segments; their interaction with machine-produced segments is simply a special case where the regularity of the line segment features reduce the complexity of the processing and the number of possible interpretations. Figure 4 shows some of the platform components including cross sections, anchor points and stroke and region boundaries.

Our goal is to interpret each non-stroke region as an extension of the set of associated segments (extrapolation) and to recover the occluded stroke contours and cross sections (reconstruction). In general, the validity of an intersection region interpretation will be based on 1) the *smoothness* of the segment pairings, 2) the *compatibility* of the stroke reconstructions with the region in question, and 3) *constraints* imposed by higher level understanding of the writing.

5.1 Smoothness

Smoothness is a local measure of the confidence that a given pair of segments i and j are portions of the same stroke, extending through a region k. In general, we define smoothness, Ψ, for each pair of segments i and j traversing a non-stroke region k as

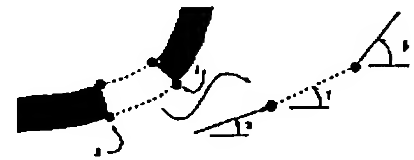


Figure 5: Smoothness parameters

$$\Psi(i,j,k) = F(\vec{\omega},\vec{P}_i,\vec{P}_j,\vec{P}_k) \quad \vec{P} = (p_1, p_2, \ldots, p_n)$$
$$= \bigodot_{m=1}^{n} \omega_m f_m(p_{im}, p_{jm}, p_{km}) \tag{1}$$

where $\vec{P}_i$, $\vec{P}_j$, and $\vec{P}_k$ are property vectors computed from the corresponding segment and region features, the f_ms are smoothness functions of individual feature properties and $\vec{\omega}$ is a weight vector. The $\bigodot$ operator is a (possibly non-linear) combination of the smoothness parameters. The smoothness computation is based on perceptual organization criteria involving the position, curvature and feature consistency of the associated segments and support from boundary features within the non-stroke region.

First we define the computation of a set of local properties within each region. At each stroke anchor point, k, we estimate the angle of the boundary measured from the horizontal axis, α_k, and the sign of its curvature, s_k, and mark its orientation. In addition, for each candidate pair of anchor points, we compute the angle, γ, of a line segment connecting them in the direction of positive orientation (Figure 5). From these parameters, various properties can be computed including:

$$\begin{aligned} bend &\Rightarrow abs(\beta - \alpha) & \Delta X &\Rightarrow t_x - t_y \\ total\ bend &\Rightarrow abs(\beta - \gamma) + abs(\gamma - \alpha) & \Delta Y &\Rightarrow t_y - t_x \\ \Delta bend &\Rightarrow (\beta - \gamma) - (\gamma - \alpha) & \|t\| &\Rightarrow \sqrt{\Delta X^2 + \Delta Y^2} \end{aligned}$$

$$\text{Compatible Curvature (CC)} \Rightarrow \begin{cases} 1 & \text{if } s_\alpha = sign(\beta) \;\&\; s_\beta = sign(\alpha) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{Region Support (RS)} \Rightarrow \begin{cases} 1 & \text{if } \exists m \text{ such that } \partial R_m = tt \\ 0 & \text{otherwise} \end{cases}$$

Given an anchor point pair, we wish to define $\bigodot$ sot that it is maximal in the smoothest case. If we assume that a stroke should form a ribbon-like region, the properties derived from the two sets of anchor point pairings can be averaged to give property values for the stroke. Similarly, the properties which are computed for more than two anchor points along the same stroke can be averaged to produce a single value.

Depending on the configuration of the incoming arcs, different smoothness criteria can be used. For example, if a merge hypothesis is generated, one possible smoothness function is

$$\Psi = \bigodot(\vec{\omega}, bend, \Delta bend, support)$$
$$= \left(\frac{\omega_1}{bend} + \frac{\omega_2}{\Delta bend} + \frac{\omega_3}{Support}\right) \tag{2}$$

which is maximum for a smooth transition to the next segment and favors the straightest path hypothesis.

5.2 Compatibility and constraints

Having defined a local pairwise smoothness criterion, it is still essential to incorporate regional (within the junction) and global (dealing with stroke continuity) components into the interpretation. The *compatibility* of the image and the reconstruction is based on the extent to which the interpretation agrees with the original image. For example, given segments i and j, the pairing is subject to verification that the width and intensity variations are within acceptable bounds. This information can be used both heuristically to constrain the possible interpretations as well as for verification of the reconstruction's consistency with the image. Additional compatibility measures include verification that the region and all of its uninterrupted boundary fragments are accounted for, and that all of the smoothness parameters are within acceptable bounds.

Handwriting *constraints* can also be incorporated into the interpretation process to help resolve ambiguity if more than one interpretation is possible. A weighting function $\phi(i,j)$ can be associated with each segment pairing i and j, and derived from handwriting constraints such as stroke continuity, temporal ordering, or global positioning.

Both the compatibility measure and the handwriting constraints are implemented as procedural rules which effect the algorithm either locally in the case of heuristics or globally during verification.

5.3 Reconstruction

In the next section we will discuss the algorithms for interpretation. Once we have derived a feasible interpretation for the region, we reconstruct the occluded segment boundaries as follows. If one boundary is visible, we hypothesize the occluded boundary by constructing a ribbon which is defined on one side by the visable boundary. The width of the stroke varies uniformly from the distance between the anchor points on one side of the region k to the distance between the anchor points on the other side of k. The cross-sections are constructed between the two boundaries so that each corresponds to the diameter of a maximal disk, thus minimizing the angular difference between the boundary normal and the cross-section angle at each end of the cross-section. Figure 6 shows some possible reconstructions.

If there are no visable boundaries or we are bridging a gap in the stroke, a cubic spline approximation is used to link the anchor points. Unless contrary evidence exists, a smooth transition from one stroke to the next is an appropriate assumption, and the smoothness properties of splines provide a reasonable reconstruction. The stroke then consists of the region enclosed by the two boundaries. Other features such as the medial axis and average intensity can easily be computed by projecting the derived representation onto the original image.

6 Region Interpretation

At the first level of discrimination, the type of intersection is weakly classified according to the number of segments involved, or equivalently the number of arcs associated with the node in the stroke graph. The most common intersection nodes involve 2, 3 or 4 arcs. If

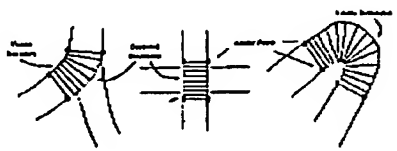
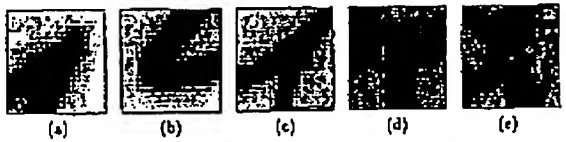


Figure 6: Partial reconstructions



(a) (b) (c) (d) (e)

Figure 7: Basic Stroke/Stroke Interactions: (a) High-curvature, (b) Corner, (c) Merge, (d) Abutment, (e) Crossing.

two arcs are present, we hypothesize either a *high-curvature point*, a corner, or explore the possibility of a boundary defect. If three arcs are present, the most likely scenarios include a merge or an abutment, and for four arcs, a crossing will be the primary type of intersection considered. More complex interactions which are also considered may appear as combinations of these fundamental junctions; they include an *extended crossing* and an *incidental contact*. For more complex intersections with a larger number of arcs, we have found that the best approach is to begin with a crossing hypothesis, and decompose it recursively to identify the best segment pairings.

6.1 High-curvature points and corners

High-curvature points are the result of a single stroke segment either having a discontinuity in curvature, or having such a large curvature that a portion of the inside boundary becomes unstable (Figure 7a). The effect of this configuration is that the stroke partially overlaps itself causing confusion in the gray level information along the inside boundary. The outer contour tends to be both visible and stable and will serve as an adequate basis for determining the stroke trajectory. To reconstruct, the outer boundary is used to anchor one side of the stroke, and the other side is defined by the endpoint of the cross-section constructed with a width consistent with the incoming segments. The inner contour is refined according to smoothness constraints.

A corner is a curvature discontinuity in the stroke trajectory and may be difficult to distinguish from a high-curvature point (Figure 7b). We use the curvatures of the approaching segments and the sharpness of the outer stable contour in an attempt to distinguish between the two. If a corner interpretation can be derived, a label is attached to the region for later analysis.

6.2 Merges and abutments

A merge is characterized by two segments which approach the same trajectory, meet and continue along the same path (Figure 7c and 7d). The third "merged" segment is often wider and/or darker than the two approaching segments since their paths are not likely to coincide precisely. An abutment is characterized by one continuous stroke with the second segment meeting it and ending.

We first attempt to identify the primary segment for the merged hypothesis, identified as the segment which forms the largest angle with the other two segments, and is possibly wider. If the difference in width of the primary and approach segments is significant, the anchor points are split, and the primary segment can be re-interpreted as a merge region or part of an intersection.

The intersection of a merge may occur over an extended region, especially if the angle between the approaching stroke segments is small (Figure 7c). For a merge, the outer contours of the segments are visible and stable and are used to reconstruct the stroke. The inner contours are in general unstable and occluded as they approach the intersection. The inner contour is recovered by constructing cross sections of the appropriate widths normal to the outer boundary. The case where the segments merge over an extended region and eventually cross is discussed in the next section

If it is found that the primary segment makes a smooth transition with only one of the other segments and their widths are similar, an abutment is considered. Higher level knowledge may again be required to resolve ambiguity.

6.3 Crossings and Extended Crossings

The most common case of intersection is a simple crossing and is characterized by two strokes intersecting at or near a 90° angle (Figure 7e). When the segments are nearly orthogonal, there will be no useful visible contours around the region. Unless there exists a large discrepancy in the properties of the approaching segments, a straight stroke segment is assumed. Alternative interpretations may be possible, and in the most general case it is useful to pass such interpretations on to higher level modules or allow reevaluation.

Two strokes which cross over an extended region may not be detectable from only local information since the intersection region may also exhibit stroke-like properties (Figure 8). This is common, for example, when the strokes meet at a low incidence angle. If two merge configurations are detected, these junctions are neighbors in the stroke graph and can be disambiguated at that point. An extended crossing may be explored in conjunction with a possible merge.

6.4 Incidental Contact

Incidental contact occurs when two strokes meet and separate without crossing. Differentiating between the situation of incidental contact and an extended intersection can be difficult at best. Since both situations occur over a long thin region, the outer boundaries may tend to be fairly smooth. It is arguable that humans rely primarily on higher-level information about the writing for disambiguation.

A simple analysis shows, however, that in order for two segments with widths a and b, $a < b$, to cross at an angle $\theta \leq 90°$, a width of

$$w = \begin{cases} \max(a,b) & \text{if } a < b\cos\theta \\ \sqrt{\left(\frac{a\sin\theta - b\cos\theta}{\sin\theta}\right)^2 + b^2} & \text{otherwise} \end{cases}$$

between the outer boundaries in the merged region must be obtained. This criterion can be used qualitatively to rule out the possibility of a crossing, as in Figures 9a and b. In other cases, the boundaries may be explored for curvature discontinuities that suggest the intersection of two separate boundaries.

(a) (b) (c)

Figure 8: Cross section representation of an extended crossing

(a) (b) (c)

Figure 9: Examples of incidental contact (a & b) and a complex intersection (c).

6.5 Complex Interactions

Complex interactions, complete stroke occlusion and interaction of segments with unknown document features fall into a class of intersections which may require non-local information to resolve. The general approach described above can be extended to attempt to interpret more complex regions which contain multiple segments and region boundary fragments. For occlusion cases it seems unlikely that the correct interpretation can be derived from local information alone. The only available local information is the intensity, which may have significant differences in some situations. This is a prime candidate for feedback from higher level modules. Knowledge of the average character size, or partial recognition results, should be considered in such situations. Similarly, when strokes intersect with unknown or non-stroke features, we preserve both the gap and filled hypotheses for higher level consideration.

Figure 10: Example cross-section representations of reconstructed strokes.

7 Results and Conclusions

The intersection regions shown in the previous examples were taken from hand-printed and hand-written address blocks scanned at 300 dpi. Figure 10 shows the cross-section representations of the reconstructions of some of these examples. The cross sections are added to the stroke platform along with their interpretations and the properties of the intersections.

Difficulties still remain in defining a single smoothness operator for each configuration. It is clear that higher level relations between strokes are necessary to resolve inherent ambiguities of the local configuration. The detection of extended crossings and incidental contacts are examples of more complex interpretations which are currently being refined in the system. Further work will include a more complete use of the boundary regions, a reconstruction in which the region boundaries are defined to sub-pixel accuracy, and a process which verifies the interpretation regionally.

Traditionally, a thin line representation of strokes or segments has been sufficient for high level analysis tasks. Unfortunately, the delineation of an accurate medial axis which represents either the writing instrument trajectory or the true mid-line of a Etc segment is not an easy task. We have provided an approach to reconstructing the original stroke segment which is based on stable properties of the document and minimal assumptions about the nature of the data. It is clear that a well-localized representation is essential for tasks such as recognition, and is one of the main difficulties for tasks such as vectorization, forms processing and text/graphics discrimination. We are in the process of applying this detailed analysis approach to stroke interpretation to several of these problems.

References

[1] D. S. Doermann. *Document Image Understanding: Integrating Recovery and Recognition*. PhD thesis, University of Maryland, College Park, 1993.

[2] D. S. Doermann and A. Rosenfeld. Recovery of temporal information from static images of handwriting. In *Proceedings of Computer Vision and Pattern Recognition*, 1992. Submitted to Internation Journal of Computer Vision.

[3] H. Nishida, T. Suzuki, and S. Mori. Thin line representation from contour representation of handprinted characters. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pages 57–68, 1991.

[4] D. Wang and S. N. Srihari. Analysis of form images. In *Proceedings of the First International Conference on Document Analysis and Recognition*, pages 181–191, 1991.

Off-Line Recognition of Large-Set Handwritten Hangul with Hidden Markov Models

Hee-Seon Park and Seong-Whan Lee

Department of Computer Science, Chungbuk National University
48 Gaeshindong, Cheongju, Chungbuk 360-763, Korea
E-mail: swlee@cbucc.chungbuk.ac.kr
Voice: +82-431-61-2263, Fax: +82-431-272-3640

ABSTRACT

In this paper, we propose an efficient off-line recognition scheme for large-set handwritten Hangul in the framework of hidden Markov model(HMM) which can model stochastically the input pattern with numerous variations. In this scheme, after extracting four kinds of regional projection contours from an input pattern by using the regional projection contour transformation, four HMMs are constructed based on the direction components of corresponding contours during the training phase. In the recognition phase, four HMMs constructed in the training phase are combined to output the final recognition result for an input pattern. For the construction of an efficient recognition system, unnecessary parameter estimation was avoided by imposing some restrictions on HMM parameters, and a fuzzy tree classifier was adopted to speed up the overall processing time.

In order to verify the effectiveness of the proposed scheme, the most frequently used 520 syllables in Korea were considered in the experiments. Experimental results indicated that the proposed scheme is very promising for the recognition of handwritten Hangul with numerous variations.

I. INTRODUCTION

Despite many attempts to build Hangul recognition systems since 1969, it is still a very challenging problem to develop a practical system[Cho92, Lee92b, Lee93a]. The main reason for this is that a Hangul recognition system should be able to classify a large set of syllables which are very similar to each other.

This research focuses on the use of stochastic models, in this case, first order hidden Markov models(HMMs), to recognize handwritten Hangul. HMMs have been widely used for automatic speech recognition[Bahl83, Baker75, Velin76, Rabin89b], and have proven successful in dealing with the statistical and sequential aspects of speech signal. Based on its success in these related areas of speech recognition, a question that arises naturally is how well these stochastic models would work on problems in character recognition. Recently, there are many on-going researches to recognize the multi-font [Anigb92, Bela191] or handwritten script[Jeng90, Kuadu86, Kuadu89, Vfont92] by using this approach. For example, Kuadu *et al.*[Kuadu89] have developed a handwritten English script recognition scheme, even to the point of using second order model. These research efforts have greatly contributed to the general understanding of the applicability of HMM to handwritten character recognition, the complexity of the parameter computation, as well as the limitation of the model.

Historically speaking, Hangul is the Korean script which was invented about 500 years ago. Each Korean script represents a syllable and is composed of several phoneme.